

ALL

Requested Patent: JP2001007822A

Title:

A PIPELINED PACKET SCHEDULER FOR HIGH SPEED OPTICAL SWITCHES ;

Abstracted Patent: EP1061763 ;

Publication Date: 2000-12-20 ;

Inventor(s): CAVENDISH DIRCEU G (US) ;

Applicant(s): NIPPON ELECTRIC CO (JP) ;

Application Number: EP20000103463 20000229 ;

Priority Number(s): US19990335908 19990618; US19990460649 19991214 ;

IPC Classification: H04Q11/00 ;

Equivalents:

ABSTRACT:

A pipelined scheduler which allows easy implementation and control and further fair scheduling among input lines of a crossbar high speed switch fabric is discussed. By means of a round-robin communication scheme, a systematically ordered sequence of visits to time slots can be obtained regardless of whether the number of scheduler modules is even or odd by framing the time axis and using a priority matrix to reserve future time slots. Further, a Carry Over Round-robin Pipelined Scheduler (CORPS) achieves scalability to a large number of ports. Moreover, CORPS achieves one scheduling decision per line per slot, by scheduling packets in future slots. It is well suited to the support of Quality of Service traffic, since the choice of the queues to be scheduled is arbitrary. CORPS limits itself to resolve, in a fair way, the contention for output ports.

【特許請求の範囲】

【請求項1】 ネットワークにおけるデータのフローを制御するスイッチにおいて、

複数の入力ポートと、

複数の出力ポートと、

前記複数の出力ポートのうちの指定出力ポートへデータを送るよう、前記複数の入力ポートのうちの特定の入力ポートをスケジューリングする複数の入力ポートスケジューリングモジュールを有するスケジューラと、からなり、

現在のスケジューリングモジュールは、

前のスケジューリングモジュールからスケジューリングメッセージを受信し、

前記現在のスケジューリングモジュールが前記指定出力ポートにアクセスしようとする将来のタイムスロットを計算し、

前記将来のタイムスロットが前記現在のスケジューリングモジュールによって既に予約されているかどうか、前記将来のタイムスロットが阻止されているかどうか、及び前記将来のタイムスロットが他のスケジューリングモジュールによって取られているかどうかに基づいて、前記将来のタイムスロットが有効かどうかを判断し、

有効な場合には、前記将来のタイムスロットを取り、前記スケジューリングメッセージに前記将来のタイムスロットが取られたことを示す情報を入れる、ことを特徴とするデータフロー制御スイッチ。

【請求項2】 前記スケジューラは、前記将来のタイムスロットが予約されている場合及び取られている場合のいずれかである時には、前記将来のタイムスロットを所定数のタイムスロットだけ前進させることを特徴とする請求項1記載のデータフロー制御スイッチ。

【請求項3】 前記複数の出力ポートのそれぞれに対して別々のキューを維持する仮想出力キューイング(VOQ)を用いて、前記複数の入力ポートを通じて入力されたデータをキューイングすることを特徴とする請求項1記載のデータフロー制御スイッチ。

【請求項4】 あるポートに対する前記仮想出力キューイングは、他のポートに対する前記仮想出力キューイングとは独立であることを特徴とする請求項3記載のデータフロー制御スイッチ。

【請求項5】 前記仮想出力キューイングのサービスレートは予測可能かつ調整可能であることを特徴とする請求項3に記載のデータフロー制御スイッチ。

【請求項6】 前記スケジューラは、重み付きラウンドロビンに基づいて、前記指定出力ポートを選択することを特徴とする請求項1記載のデータフロー制御スイッチ。

【請求項7】 複数の入力ポートスケジューリングモジュールを有するスイッチの複数の入力ポートに到着する入力信号を当該スイッチの複数の出力ポートに送るよう

にスケジューリングする方法において、

a) 現在のスケジューリングモジュールが、前のスケジューリングモジュールからスケジューリングメッセージを受信するステップと、

b) 前記現在のスケジューリングモジュールが、前記複数の出力ポートのうちの1つにアクセスしようとする将来のタイムスロットを計算するステップと、

c) 前記複数の出力ポートのうちの1つを前記将来のタイムスロットでの送信用にスケジューリングするよう

に選択するステップと、

d) 前記将来のタイムスロットが前記現在のスケジューリングモジュールによって既に予約されているかどうかを判断するステップと、

e) 前記将来のタイムスロットが前記現在のスケジューリングモジュールによって予約されていない場合には、前記将来のタイムスロットが阻止されているかどうかを判断するステップと、

f) 前記将来のタイムスロットが阻止されていない場合には、前記将来のタイムスロットが他のスケジューリングモジュールによって既に取られているかどうかを判断するステップと、

g) 前記将来のタイムスロットが、他のスケジューリングモジュールによって既に取られている場合及び前記現在のスケジューリングモジュールによって既に予約されている場合のいずれかの場合には、前記スケジューリングメッセージから繰越し動作が既に開始されているかどうかを判断するステップと、

h) 前記繰越し動作が既に開始されている場合には、前記将来のタイムスロットを阻止状態に設定して前記ステップ(d)に戻るステップと、

i) 前記繰越し動作が開始されていない場合には、前記将来のタイムスロットを所定数のタイムスロットだけ前進させ、繰越しフラグをセットしてステップ(d)に戻るステップと、

j) 前記将来のタイムスロットが他のスケジューリングモジュールによって取られていない場合には、前記将来のタイムスロットを取り、前記将来のタイムスロットが取られたことを示す情報を前記スケジューリングメッセージに入れるステップと、

k) 前記スケジューリングメッセージを次のスケジューリングモジュールに渡すステップと、からなることを特徴とするスケジューリング方法。

【請求項8】 前記複数の入力ポートを通じて入力したデータは、各出力ポートに対して別々のキューを維持する仮想出力キューイングを用いてキューイングされることを特徴とする請求項7記載の方法。

【請求項9】 あるポートに対する前記仮想出力キューイングは、他のポートに対する前記仮想出力キューイングとは独立であることを特徴とする請求項8記載の方法。

【請求項10】 前記仮想出力キューイングのサービスレートは予測可能かつ調整可能であることを特徴とする請求項8記載の方法。

【請求項11】 前記スケジューラは、重み付きラウンドロビンに基づいて、前記指定出力ポートを選択することを特徴とする請求項7記載の方法。

【請求項12】 ネットワークにおけるデータのフローを制御するスイッチにおいて、

複数の入力ポートと、

複数の出力ポートと、

前記複数の出力ポートのうちの指定出力ポートへデータを送るよう、前記複数の入力ポートのうちの特定の入力ポートをスケジューリングするN個の複数の入力ポートスケジューリングモジュールを有するスケジューラと、

から構成され、

前記スケジューラは、

前記各入力ポートスケジューリングモジュールがリング状に接続され、

タイムスロット単位に、

各入力ポートスケジューリングモジュールが、前段のスケジューリングモジュールから、ある予約タイムスロットの予約状況情報を受信し、

各入力ポートスケジューリングモジュールが、その予約タイムスロットにおける当該入力ポートスケジューリングモジュールからのバケット送出手続き可否を決定し、各入力ポートスケジューリングモジュールが、前段のスケジューリングモジュールから受信した予約状況情報に、自スケジューリングモジュールの予約結果を反映させて、次段のスケジューリングモジュールに送信する、ことを特徴とするデータフロー制御スイッチ。

【請求項13】 複数の入力スケジューリングモジュールを有するバケットスイッチの入力ポートと出力ポートの接続状態を決定し接続を予約する（以下、スケジューリングという。）方法において、

N個のタイムスロットを単位とするフレームを定義して、前記フレーム時間内で、前記フレームの次フレーム中のN個のタイムスロットでのスケジューリングを行う、

ことを特徴とするスケジューリング方法。

【請求項14】 前記スケジューリング方法は、

a) 現在のスケジュールモジュールが、前のスケジュールモジュールからスケジューリングメッセージを受信するステップと、

b) 前記現在のスケジュールモジュールが、前記複数の出力ポートのうちの1つにアクセスしようとする将来のタイムスロットを予め次フレーム内の特定のタイムスロットに決定するステップと、

c) 前記複数の出力ポートのうちの1つを前記将来のタイムスロットでの送信用にスケジューリングするように

選択するステップと、

d) 前記将来のタイムスロットが他のスケジュールモジュールによって既に取られているかどうかを判断するステップと、

e) 前記将来のタイムスロットが他のスケジュールモジュールによって取られていない場合には、前記将来のタイムスロットを取り、前記将来のタイムスロットが取られたことを示す情報を前記スケジューリングメッセージに入れるステップと、

f) 前記スケジューリングメッセージを次のスケジュールモジュールに渡すステップと、

からなることを特徴とする請求項13記載のスケジューリング方法。

【請求項15】 前記スケジューリング方法は、タイムスロットでのスケジューリング決定過程（接続決定過程）の観点から見た場合、前記複数の接続決定過程が、

フレームの先頭で同時に開始され、

フレーム内で同時にパイプライン処理により進行し、

フレームの末端で同時に完了する、ことを特徴とする請求項13記載のスケジューリング方法。

【請求項16】 前記スケジューリング方法は、

前記入力ポートスケジューリングモジュールが、フレームの先頭で同時に開始する前記各接続決定過程において、次のフレーム内の各々異なる予約タイムスロットを対象として処理を開始することを特徴とする請求項13記載のスケジューリング方法。

【請求項17】 前記スケジューリング方法は、

ある与えられた将来のタイムスロットにおけるN個の入力ポートスケジューリングモジュールの規則的な巡回順序を定義するN×Nマトリクスを参照することによって、現在のフレームにおける入力信号が次のフレームでどの出力ポートへ送出手続きされるかを決定することを特徴とする請求項13記載のスケジューリング方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ネットワークによるデータのフローを制御するネットワークシステムおよびスイッチに係り、特に、大容量スイッチを通じてのデータフローを管理するスケジューラに関する。

【0002】

【従来の技術】入力キュースイッチアーキテクチャは、常に高速交換システムの有力な選択肢であった。それは主に、入力バッファのメモリアクセス速度が、全交換容量とともにスケールするのではなく、単一の入線の速度とともにスケールするからである。しかし、入力バッファ型スイッチは、HOL(head-of-line)ブロッキングの問題があることが以前から知られており、これにより全スループットは58.6%という理論限界に制限される(M. J. Karol, M. G. Hluchyj, S. P. Morgan, "Input

Versus Output Queuing on a Space-Division Packet Switch", IEEE Transactions on Communications, Vol. C OM-35, No. 12, pp. 1347-1356, Dec. 1987, 参照)。

【0003】最近になって、入力スイッチのHOLブッキングの問題を克服するために、仮想出力キューイング(VOQ: Virtual Output Queuing)という入力キューイング方式が提案された(Y. Tamir and G. Frazier, "High Performance Multi-queue Buffers for VLSI Communication Switches", Proceedings of 15th Ann. Symp. on Comp. Arch., pp. 343-354, June 1988, および、T. Anderson, S. Owicki, J. Saxe, C. Thacker, "High Speed Switch Scheduling for Local Area Networks", ACM Transactions on Computer Systems, pp. 319-352, Nov. 1993, 参照)。その考え方は、スイッチの各出力ポートごとに別々のキューを設け、空き出力ポート宛のバケットが、別のポートに対する競合により進めない先頭バケットによってサービスをブロックされる可能性がなくなるようにするというものである。この場合、 $N \times N$ スイッチは入力ポートごとに N 個のキュー、すなわち、 N^2 個のキューを有する。他の研究者によって議論されているように(A. Mekittikul, N. McKeown, "A Practical Scheduling Algorithm to Achieve 100% Through-put in Input-Queued Switches", Proceedings of Infocom98, April 1998, 参照)、VOQ法をさらに研究することによって、高性能のスケジューラ的设计により実に100%のスループットが達成可能であることが示されている。

【0004】従って、VOQ入力バッファ型スイッチのスケジューラは高速入力バッファ型スイッチの重要な設計ポイントのうちの1つとなる。VOQの場合、スケジューラは、通常の先入力先出力(FIFO)入力キューイングアーキテクチャの場合よりも、バックログのある入力ポートから出力ポートへバケットを交換するのにはるかに多くの選択肢を有する。バックログのある入力ポートのうちで、あらゆる入出力ポート対を選択することができる。

【0005】このようなスケジューラに対する研究のほとんどは以下のように分類することができる。集中スケジューラは、スケジューラが N^2 個のすべてのVOQに関する情報を有する単一のエンティティであって、バケットスロットごとにすべての可能な入出力ポート対に関するスケジューリング決定を行うものである(例えば、A. Mekittikul, N. McKeown, "A Practical Scheduling Algorithm to Achieve 100% Through-put in Input-Queued Switches", Proceedings of Infocom98, April 1998, 参照)。

【0006】他方、分散スケジューラは、スケジューラがいくつかの機能ブロック(通常は入力あるいは出力ポート当たり1又は2個のブロック、あるいは、入出力クロスポイント当たり1ブロック)に分割されたものであ

る(例えば、N. McKeown, H. Izzard, A. Mekittikul, W. Ellersick, M. Horowitz, "The Tiny Tera: A Packet Switch Core", IEEE Micro, Jan/Feb 1997, pp. 26-32, および、Y. Tamir and H-C Chi, "Symmetric Crossbar Arbiters for VLSI Communication Switches", IEEE Transactions on Parallel and Distributed Systems, Vol. 4, No. 1, pp. 13-27, 1993, 参照)。

【0007】図1は、集中スケジューラの一例を示す機能的ブロック図である。集中スケジューラは、スケジューリング決定を行う前に、 N^2 個の情報にアクセスする必要がある。このようなスケジューラは、一般に、スケジューラを実装するハードウェアがスイッチラインの数 N に強く依存するという意味で、スケーラブルではない。

【0008】分散スケジューラは、ハードウェアに対してスイッチポート数への依存性をより少なくする可能性を有する。しかし、これまで提案されているものは依然として、個々のバケットスロットに対するスケジューリング決定を行うことができる前に、 N^2 個のすべてのキューに関する情報を提供する通信メカニズムを必要とする。この通信は、並列に(SLI Pスケジューラの場合のように、N. McKeown, M. Izzard, A. Mekittikul, W. Ellersick, M. Horowitz, "The Tiny Tera: A Packet Switch Core", IEEE Micro, Jan/Feb 1997, pp. 26-32, 参照)、あるいは、ラウンドロビン方式で(Y. Tamir and H-C Chi, "Symmetric Crossbar Arbiters for VLSI Communication Switches", IEEE Transactions on Parallel and Distributed Systems, Vol. 4, No. 1, pp. 13-27, 1993, 参照)行うことが可能である。

【0009】図2(A)及び(B)は、並列方式及びラウンドロビン方式のアーキテクチャをそれぞれ示す模式図である。図2(A)における並列通信アーキテクチャでは、各ブロックがスイッチのサイズに陽に依存してしまう。各ブロックが N^2 個のメッセージを受け取らなければならないからである。図2(B)におけるラウンドロビンアーキテクチャはこの問題点を克服するが、別の問題点を生じる。すなわち、すべての出力ポートに関するスケジューリング決定を達成するためには、メッセージ受渡しは、単一のバケットスロット中に1ラウンドを完了しなければならないという点である。これは、スケジューリング決定よりも少なくとも N 倍高速なメッセージ処理を必要とする。

【0010】さらに最近になって、ラウンドロビン・グリーディ・スケジューラ(RRGS: Round-Robin Greedy Scheduler)が提案された。これは、メッセージバッシング(受渡し)に基づくスケジューラであり、各入力ポートがスケジューリング決定を行い、この情報をラウンドロビン方式で次のポートに渡すものである(本出願人による特願平11-172584号を参照)。メッセージ受渡し速度要求条件を緩和するために、RRGSは

パイプライン機能を導入している。入力ポートは、十分将来のスロットに関するスケジューリング決定を行い、メッセージ受渡しメカニズムがこの情報を他の入力ポートに広めるのに十分な時間があるようにする。RRGSは高速なスケジューリングを実現することができる。

【0011】まず、一般的なパイプライン型スケジューラのアーキテクチャについて説明する。図3は入力バッファスイッチアーキテクチャを例示する模式図である。スイッチアーキテクチャに関して、スケジューリングは、純粋なノンブロッキング $N \times N$ クロスバスイッチに適用されると仮定する。また、仮想出力キュー(VOQ)を用いてHOLブロッキング問題に対処すると仮定する。

【0012】さらに、固定サイズパケットおよび一樣リンク速度を仮定する。時間はスロット化される。1つのスロットは、出力リンクによる1パケットの送信にかかる時間として定義される。出力ポート競合が存在しない場合、ノンブロッキングクロスバは、タイムスロット当たり N パケットまでを交換することができる。スケジューラの基本的な仕事は、スロットごとに、空でない N^2 個のVOQキューのうちのいずれが出力ポートにアクセスすることができるかを判断することである。効率のために、スケジューラは、1タイムスロット内でバックログのあるキューの間のすべての競合を解決しなければならない。

【0013】ライン速度が増大すると、スケジューリングアルゴリズムが大容量スイッチにもスケーラブルであることが重要となる。従って、分散アーキテクチャが有力であると思われる。分散アーキテクチャでは、高速スイッチにおいてパケットスケジューリングに要求されるきつい処理時間が緩和されるからである。例えば、10 Gbit/sのライン速度の 16×16 ポートスイッチで、スケジューリング決定は、各パケット送信時に行わなければならない。424ビットのATMセルに対して42 nsである。シーケンシャルスケジューラを使用する場合、各決定は、 16×16 スイッチでは0.16 ns未満で行わなければならない。 N^2 個の決定をしなければならないからである。光コアを使用する場合、光コアの全交換帯域幅要求条件をそのままにして、電子ハードウェアをポートごとに分散することには意味がある。さらに、分散スケジューラは、当然、任意のライン数にあわせてスケールされる。図4にそのようなスケジューラを例示する。

【0014】図4において、各クロスバ入力ポートは、入力ポートスケジューラモジュール(SM: Scheduler Module)を有する。各SMは、個別のIDであるSM-IDを有する。ライン数とのスケーラビリティを維持するために、SMは、隣の1個のSMとのみ通信することが許される。これにより、SMハードウェアブロックは任意の $N \times N$ クロスバファブリックで使用可能であるこ

とが保証される。SM通信チェーンが図4に示されている。これは、タイムスロット、スロット所有権、および出力ポート予約のようなスケジューリング情報を通信するために使用される。クロスバモジュールとSMとの間の唯一の相互作用は、グローバルクロックを通じてのものである。これは、あらゆるSMに、どのスロットが現在のシステムタイムスロット(CTS: Current system Time Slot)であるかということと、CTSで交換される入出力ポート対に関する現在の決定テーブル(図示せず)とを知らせる。これは、スケジューラによって書き込まれ、クロスバファブリックによって読み出されるグローバルメモリによって実現することが可能である。

【0015】タイムスロットごとに、各SMは、アクセス要求先の出力ポートに関して完全な選択の自由があると仮定される。同様の選択をするSMどうしは「コリジョン」(衝突)を生じ、これは、与えられたスロットに対するグローバルスケジューリングパターンを決定する前に解決する必要がある。SMが、他のすべての要求に関する現在の情報を有することになる場合、通信チェーンは、スケジューリング決定の速度よりも N 倍速い速度で動作しなければならない。すなわち、SMは、1つのスケジューリング決定を行う前に、 N 個のメッセージを受信することができなければならない。

【0016】SMハードウェアの速度をライン速度とともにスケーラブルに保つために、 N ルックアヘッド(先読み)スケジューリング方式を使用することが可能である。すなわち、各SMは、現在のスロットの少なくとも N スロット先のタイムスロットに関してスケジューリング決定をすることになる。この機能により、SMは、スケジューリング決定をする前に、同じタイムスロットに対してなされている他のスケジューリング決定に関して知っていることが保証される。さらに、この機能は、通信チェーンを入力ライン速度の N 倍に高速化する必要がない。RRGSは上記のような、分散スケジューリング、パイプラインスケジューリングの特徴と、 N ルックアヘッド(先読み)スケジューリングの特徴を備えている。

【0017】図5は、 4×4 クロスバスイッチを用いた場合のRRGSスケジューリングの一例を示すタイムチャートである。図5では、4個のSM1~SM4と、それらの入力出力ポートを選択するタイムスロットT6、T7・・・との関係が示されている。

【0018】図5において、例えばタイムスロットT5で、SM1はタイムスロットT10で送信を行うための出力ポートの選択(スケジューリング)を行い、SM3はタイムスロットT9におけるスケジューリングを行っている。また、次のタイムスロットT6では、SM1はタイムスロットT8におけるスケジューリングを行っている。以下同様である。

【0019】上記のように各SMがスケジューリングを

行い、その結果を次段のSMに転送することによって、あらゆるSMが、既にスケジューリングされたポートに関する情報を適時に得ることが保証される。あるSMが、前の「訪問者(visitor)」(即ち、1タイムスロット前のSM)によって既に選ばれた出力ポートを選ぶことを避ければ、コリジョンを完全に回避することができる。

【0020】

【発明が解決しようとする課題】しかしながら、RRGSでは、一つのSMが予約を行っていくタイムスロットの巡回順序が複雑になる。図6は、図5を個々のSMのタイムスロット巡回順序に着目して表現したタイムチャートである。例えば、SM1について見ると、タイムスロット巡回順序は、T10、T8、T11、T9・・・となり、時系列的あるいは逆時系列的な一定の規則的順序にはなっていない。これはRRGSの実装及び制御が複雑になるという問題を示している。

【0021】さらに、前記特願平11-172584号に示されるように、RRGSは、SMが偶数の場合と奇数の場合でスケジューリング動作が異なる。これは、SMを追加する際に制御を変更しなければならないことを示しており、実装及び制御が複雑になるという問題がある。

【0022】また、RRGSでは、SMが、まだ選ばれていない出力ポートを選ぶように制限されるため、VOQサービスレートを予測が困難になる。さらに、重大な公平性の問題が生じる。例えば、図4において、SM#1とSM#2は与えられた出力ポートのキューにコンスタントにバックログがあり、他のSMの対応するキューは空であるとする。この場合、SM#1は、図5にて定義される巡回順序においてSM#2の前に4スロットのうちの3スロットを訪れるため、4スロットのうち3スロットはSM#1によって取られることになる(前記特願平11-172584号を参照)。

【0023】このように、上記のRRGSスケジューラは高速なスケジューリングを実現することが可能であるが、実装及び制御が複雑になるという問題がある。また、予測可能かつ調整可能なサービスレートを実現できない。また、上述したようにVOQのいくつかが他のVOQの状態によりスケジューリングを妨げられるという公平性の問題もある。

【0024】本発明の目的は、RRGSの実装及び制御の複雑さを解消した簡易なスケジューラの基本方式を提供することにある。

【0025】本発明の他の目的は、VOQキューがスケジューリングを行う際の完全な選択の自由を可能にする大容量スイッチのためのスケジューラを提供することにある。

【0026】本発明のさらに他の目的は、VOQサービスレートを予測可能かつ調整可能にするスケジューラを

提供することにある。

【0027】本発明のその他の目的は、いずれのVOQも他のVOQの状態に拘わらず同じ確率でスケジューリングされるという意味で公平であるようなスケジューラを提供することにある。

【0028】スケジューラ設計におけるもう1つの制約は、次にスケジューリングされる与えられた入線に属するN個のVOQのうちのどのVOQを、スケジューラの制御から外すかの決定である。換言すれば、入力ポートごとに、どの出力ポートが次にスケジューリングされるかを、ある外部エンティティが全く自由に決定することである。この要請は将来のサービス品質(QoS)のサポートにとって重要である。これによりVOQの予測可能なサービスレートをより予測可能にする最大スループットを低下させる可能性があることは明らかである。しかしながら、これは重要な点である。スイッチ全体のスループットの最大化は、一部のキューの枯渇、ひいては、それらのキューに関連するフローも妨げられる可能性があるからである。

【0029】

【課題を解決するための手段】本発明の第1の観点によれば、ネットワークにおけるデータのフローを制御するスイッチは、複数の入力ポートと、複数の出力ポートと、複数の入力ポートスケジューラモジュールを有するスケジューラとを有する。各スケジューラモジュールは、前記複数の出力ポートのうちの指定された出力ポートへデータを送るように、前記複数の入力ポートのうちの特定の入力ポートをスケジューリングする。スケジューラモジュールは、モジュール間でスケジューリングメッセージを受け渡し、各スケジューラモジュールは、当該スケジューラモジュールが指定出力ポートにアクセスしようとする将来のタイムスロットを計算する。スケジューラモジュールは、更に、前記将来のタイムスロットが当該スケジューラモジュールによって現在予約されているかどうか、前記将来のタイムスロットが阻止されているかどうか、及び前記将来のタイムスロットが他のスケジューラモジュールによって取られているかどうかに基づいて、前記将来のタイムスロットが有効かどうかを判断する。有効な場合、スケジューラモジュールは前記将来のタイムスロットを取り、スケジューリングメッセージに前記将来のタイムスロットが取られたことを示す情報を入れる。

【0030】スイッチのスケジューラは、前記将来のタイムスロットが予約されているとき又は取られているときに、前記将来のタイムスロットを所定数のタイムスロットだけ前進させる。

【0031】スイッチは、前記出力ポートのそれぞれに対して別々のキューを維持する仮想出力キューイングを用いて、前記入力ポートを通じてデータ入力をキューイングする。あるいは、個々のポートに対する仮想出力キ

ューイングは、他のポートに対する仮想出力キューイングとは独立であることも可能である。さらに、スイッチは予測可能かつ調整可能な仮想出力キューイングのサービスレートを有する。また、スイッチスケジューラは重み付きラウンドロビンに基づいて指定出力ポートを選択する。

【0032】本発明の第2の観点によれば、スイッチの複数の入力ポートに到着する入力バケットをスイッチの複数の出力ポートに送るようにスケジューリングする方法が提供される。ここで、スケジューラは、複数の入力ポートスケジューリングモジュールを有する。この方法は、
 a) 現スケジューリングモジュールが、前のスケジューリングモジュールからスケジューリングメッセージを受信するステップと、
 b) 前記現スケジューリングモジュールが、前記複数の出力ポートのうちの1つにアクセスしようとする将来のタイムスロットを計算するステップと、
 c) 前記将来のタイムスロットにおける送信用にスケジューリングするように前記複数の出力ポートのうちの1つを選択するステップと、
 d) 前記将来のタイムスロットが前記現スケジューリングモジュールによって既に予約されているかどうかを判断するステップと、
 e) 前記将来のタイムスロットが前記現スケジューリングモジュールによって予約されていない場合、前記将来のタイムスロットが阻止されているかどうかを判断するステップと、
 f) 前記将来のタイムスロットが阻止されていない場合、前記将来のタイムスロットが他のスケジューリングモジュールによって既に取られているかどうかを判断するステップと、
 g) 前記将来のタイムスロットが、他のスケジューリングモジュールによって既に取られているか又は前記現スケジューリングモジュールによって既に予約されている場合、前記スケジューリングメッセージから、繰越し動作が既に開始されているかどうかを判断するステップと、
 h) 前記繰越し動作が既に開始されている場合、前記将来のタイムスロットを阻止状態に設定してステップdに戻るステップと、
 i) 前記繰越し動作が開始されていない場合、前記将来のタイムスロットを所定数のタイムスロットだけ前進させ、繰越しフラグをセットして、ステップdに戻るステップと、
 j) 前記将来のタイムスロットが他のスケジューリングモジュールによって取られていない場合には前記将来のタイムスロットを取り、前記将来のタイムスロットが取られたことを示す情報を前記スケジューリングメッセージに入れるステップと、
 k) 前記スケジューリングメッセージを次のスケジューリングモジュールに渡すステップと、からなる。

【0033】複数の入力ポートを通じて入力するデータは、各出力ポートに対して別々のキューを維持する仮想出力キューイングを用いてキューイングされる。個々のポートに対する仮想出力キューイングは他のポートに対する前記仮想出力キューイングとは独立である。また、前記仮想出力キューイングのサービスレートは予測可能かつ調整可能である。スケジューラは、重み付きラウンドロビンに基づいて、前記指定出力ポートを選択する。

【0034】本発明の第3の観点によれば、ネットワークにおけるデータのフローを制御するスイッチは、複数の入力ポートと、複数の出力ポートと、前記複数の出力ポートのうちの指定出力ポートへデータを送るように、前記複数の入力ポートのうちの特定の入力ポートをスケジューリングするN個の複数の入力ポートスケジューリングモジュールを有するスケジューラと、から構成され、前記スケジューラは、前記各入力ポートスケジューリングモジュールがリング状に接続され、タイムスロット単位に、各入力ポートスケジューリングモジュールが、前段のスケジューリングモジュールから、ある予約タイムスロットの予約状況情報を受信し、各入力ポートスケジューリングモジュールが、その予約タイムスロットにおける当該入力ポートスケジューリングモジュールからのバケット送出予約可否を決定し、各入力ポートスケジューリングモジュールが、前段のスケジューリングモジュールから受信した予約状況情報に、自スケジューリングモジュールの予約結果を反映させて、次段のスケジューリングモジュールに送信する。

【0035】複数の入力スケジューリングモジュールを有するバケットスイッチの入力ポートと出力ポートの接続状態を決定し接続を予約する（以下、スケジューリングという。）方法は、N個のタイムスロットを単位とするフレームを定義して、前記フレーム時間内で、前記フレームの次フレーム中のN個のタイムスロットでのスケジューリングを行うことを特徴とする。

【0036】上記スケジューリング方法は、

a) 現在のスケジューリングモジュールが、前のスケジューリングモジュールからスケジューリングメッセージを受信するステップと、
 b) 前記現在のスケジューリングモジュールが、前記複数の出力ポートのうちの1つにアクセスしようとする将来のタイムスロットを予め次フレーム内の特定のタイムスロットに決定するステップと、
 c) 前記複数の出力ポートのうちの1つを前記将来のタイムスロットでの送信用にスケジューリングするように選択するステップと、
 d) 前記将来のタイムスロットが他のスケジューリングモジュールによって既に取られているかどうかを判断するステップと、
 e) 前記将来のタイムスロットが他のスケジューリングモジュールによって取られていない場合には、前記将来のタ

タイムスロットを取り、前記将来のタイムスロットが取られたことを示す情報を前記スケジューリングメッセージに入れるステップと、

f) 前記スケジューリングメッセージを次のスケジューリングモジュールに渡すステップと、からなる。

【0037】さらに、上記スケジューリング方法は、タイムスロットでのスケジューリング決定過程(接続決定過程)の観点から見た場合、前記複数の接続決定過程が、フレームの先頭で同時に開始され、フレーム内で同時にパイプライン処理により進行し、フレームの末端で同時に完了する。

【0038】また、このスケジューリング方法は、前記入力ポートスケジューリングモジュールが、フレームの先頭で同時に開始する前記各接続決定過程において、次のフレーム内の各々異なる予約タイムスロットを対象として処理を開始する。

【0039】上記スケジューリング方法は、ある与えられた将来のタイムスロットにおけるN個の入力ポートスケジューリングモジュールの規則的な巡回順序を定義する $N \times N$ マトリクスを参照することによって、現在のフレームにおける入力信号が次のフレームでどの出力ポートへ送出されるかを決定する。

【0040】

【発明の実施の形態】本発明によるキャリーオーバー(繰越し)ラウンドロビン・パイプライン・スケジューラ(Carry Over Round-robin Pipelined Scheduler、以下CORPSという。)は、高速クロスバファブリックに対する公平なスケジューラであり、従来技術のスケジューラの問題点を解決するものである。CORPSは、高速スイッチファブリックのライン速度およびライン数の双方に関するスケラビリティ性を有する。ライン数に関するスケラビリティのために、メッセージ受渡しを有する分散アーキテクチャが選択される。更に、RRGSと同様に、メッセージ処理要求条件をライン速度とともにスケラブルに維持するために、パイプラインアーキテクチャが用いられる。

【0041】さらに、本発明によれば、時間軸を単にN個の連続するスロット列であるスロットフレームに分割して、時間をフレームの列とみなす。競合するスケジューラモジュールSM間のコリジョンを解決する基準を設定するために、優先マトリクスを使用する。 $N \times N$ 優先マトリクスは、将来の与えられたタイムスロットをSMが巡回する順序を定義するマトリクスである。マトリクスの行は現在のフレーム(現在のシステムスロットを含むフレーム)内のスロットをインデックス付けし、マトリクスの列は次に訪れるフレーム内のスロットをインデックス付けしている。マトリクスの要素は、どのSMが、列インデックスによって示される次フレーム内のスロットを「訪れる(visit)」べきかを指定する。

【0042】図7は 4×4 優先マトリクスを示す図であ

り、図8は、図7に示したマトリクスに対するパイプライン化されたタイムスロット巡回順序を例示している。パイプライン化された決定プロセスは、優先マトリクスの使用に既に含まれていることに注意すべきである。例えば、システムの現在のタイムスロットが現フレームの第2スロットであるとき、SM#1が次フレームの第4スロットに関するスケジューリング決定をしている間に、SM#3は次フレームの第2スロットに関するスケジューリング決定をしている。

【0043】優先マトリクスを使用することにより時間軸がフレーム化され、タイムスロット巡回順序が規則的になる。例えば、フレームF1における各SMの動作に着目すると、各SMのスケジューリング決定過程は、フレームF1の先頭で同時に開始され、スケジューリング決定を行うタイムスロットの巡回順序は $T8 \rightarrow T7 \rightarrow T6 \rightarrow T5 \rightarrow T8$ となり、フレームF1の末端で同時に完了する。これは、図6に示したRRGSにおけるタイムスロット巡回順序($T10, T8, T11, T9 \dots$)と比較して規則的になっている。このためSMの実装及び制御が容易になる。更に、 $N \times N$ 優先マトリクスはSMの個数に関して偶奇の区別なく、同一規則にてタイムスロットの巡回順序を定義する。

【0044】図9は 5×5 優先マトリクスを示す図であり、図10は、図9に示したマトリクスに対するパイプライン化されたタイムスロット巡回順序を例示している。フレームサイズは5タイムスロットとなるが、 $N=4$ の場合と同様に、タイムスロット巡回順序は規則的なものとなる。図8と同様に、フレームF1における各SMの動作に着目すると、各SMのスケジューリング決定過程は、フレームF1の先頭で同時に開始され、スケジューリング決定を行うタイムスロットの巡回順序が $T10 \rightarrow T9 \rightarrow T8 \rightarrow T7 \rightarrow T6 \rightarrow T10$ となり、フレームF1の末端で同時に完了する。SMの個数に関して偶奇の区別なく同一規則にて $N \times N$ 優先マトリクスを規定できるため、RRGSと比較してSMの実装及び制御が容易になる。

【0045】 $N \times N$ 優先マトリクスは、通信チェーンメッセージ受渡しの同じ方向に、SMの列を回転させることによって生成される。これにより、あらゆるSMが、既にスケジューリングされたポートに関する情報を適時に得ることが保証される。あるSMが、前の「訪問者(visitor)」(即ち、1タイムスロット前のSM)によって既に選ばれた出力ポートを選ぶことを避ければ、コリジョンを完全に回避することができる。

【0046】以上のように、時間軸をフレーム化して優先マトリクスを用いた予約を行うことにより、SMの個数に関して偶奇の区別なくタイムスロット巡回順序が規則的なものとなり、SMの実装及び制御が容易になる。

【0047】更に、本発明では、分散アーキテクチャ及びメッセージ受渡し方式を維持しつつ公平性を提供する

ために、繰越し(キャリーオーバー)動作を導入する。この考え方は、あるスケジューラモジュールSMaが、それに先行するスケジューラモジュールSMによって所望の出力ポートが既に予約されているスロットを訪れるときに、処理しようとしたスロットから将来にNスロット分だけそのポートのスケジューリング試行を繰り越すというものである。もし当該スロットが同じ出力ポートに取られていることが分かれば、SMは所望の出力ポートがまだ取られていないスロットを見つけるまで更にNスロット先に進む。

【0048】図11は、複数のスケジューラモジュールSMの間での繰越し動作を例示する説明図である。繰越し動作は、与えられたタイムスロットにおいて「衝突している」SMの個数に依存してNフレームまで広げることができる。繰越し動作によって影響されるスロットは、コリジョン(衝突)を解決するために取られるスロットの集合(以下、コリジョン解決セットという。)とみなすことができる。なお、繰越し動作を受けたスロットは、後続するフレームですべてのSMにより再び訪問されるであろう。従って、繰越し動作によって取られるスロットは、潜在的に新たなコリジョンを受け、コリジョン解決セットの重畳を引き起こす可能性がある。これは、複数のコリジョンを解決するために N^2 個のフレーム、すなわち全部で N^3 個のスロットを必要とする可能性がある。

【0049】システムのメモリ要求条件を緩和すると共にスケジューリング遅延を短縮するために、繰越し動作によって影響されるフレームの個数は、繰越し動作を実行したSMが、当該コリジョンを解決するために取られた複数のスロットにわたって同じ出力ポートに対して他のスケジューリングをしないように制限される。換言すれば、1つのスロットが、同時に複数のコリジョンを解決するには使用されない。

【0050】例えば、SMaが、与えられたポートpによって取られたスロットmを見つけ、これにより繰越し動作がトリガされると仮定する。この繰越し動作の結果としてSMaによって予約されたスロットをmxとする。スロットmn($1 \leq n < x$)のいずれも、同じポートpについてはSMaにとって利用不可(阻止、ブロッキング)となる。従って、この阻止機能は、与えられたスロットに関する複数のコリジョンが禁止されることを保証する。

【0051】CORPSスケジューリングアルゴリズムについて以下で説明する。まず、通信チェーンで渡されるメッセージと、スケジューリング決定が記録されるSMデータベースについて説明し、その後、アルゴリズムの流れについて説明する。

【0052】各セルスロットで、チェーン内のあるSMから次のSMに渡されるスケジューリング決定要素のベクトルを定義する。Sメッセージは、ただだか最後のN

個のセルスロットでなされたスケジューリング決定のスケジューリング要素(scheduling element、以下SEと記す。)を含む。すなわち、Sメッセージは、ただだかN個のSEを有する。Sメッセージは以下のフォーマットを有する。

【0053】図12は、Sメッセージのフォーマットを示す図である。同図において、Sメッセージの各スケジューリング要素SEは、存続時間(TTL: Time To Live)、タイムスロットID(TSI: Time Slot ID)、SM-ID、及び出力ポートID(OUT: Output Port ID)からなる。

- ・存続時間(TTL)は、当該SEを生成したSMによって最初にNにセットされる。

- ・タイムスロットID(TSI)は、現在のTS(タイムスロット)からスロットがスケジューリングされるまでのスロット数として定義される、スケジューリングされるスロットのIDである。

- ・SM-IDは、スケジューリング予約をした入力ポートスケジューリングモジュールのIDである。

- ・出力ポートID(OUT)は、スケジューリングされる出力ポートのIDである。

【0054】スロットの最初に、各SMは先行するSMからSメッセージを受信する。これは、最後のN個のスロットに付けられたSEを含む。あらゆるSMは、タイムスロット当たりただだか1回のスケジューリング決定を行う。SMpがスケジューリング決定を行う場合、SMpは以下の内容を有する新しいSEを作成する。

【0055】・TTL=N

- ・TSI=現在のスロットから選択されたタイムスロットまでの(それを含む)スロットの個数m

- ・SM-ID=p

- ・OUT=タイムスロット(CTS+m)におけるパケットが入力ポートpから出力ポートqに交換されるような所望の出力ポートq。

【0056】スケジューリング決定にかかわらず、各SMは、次のSMにメッセージを渡す前に、Sメッセージ内の他のすべてのSEのTTLをデクリメントし、TTL=0のSEを廃棄する。

【0057】各SMは、(N+1)N個の位置を含むメモリアレイSCを有する。最初のN個の位置は、クロスバスイッチモジュールによって読み出される現フレームのスケジューリング決定を記録する。これらの位置は、すべてのSMの間で、現フレームに関する同一の情報を有し、いくつかの方法でクロスバコントローラによってアクセスされ得る。厳密に言えば、SMは、この情報を保持する必要はない。残りの N^2 個の位置は、将来のスケジューリング決定を記録するために使用される。メモリアレイは以下のフォーマットを有する。

【0058】図13に示すように、以下のフィールドが定義される。

・タイムスロットID: SCアレイへのインデクスである。これは、SC位置がスケジューリング情報を保持するタイムスロットIDを与える。これは、クロスバモジュールによって提供されるグローバルクロックと同期する。このフィールドは、グローバルクロックが進行すると共にラップアラウンドする。

【0059】・阻止 (Blockage): これはSMがスケジューリング予約をすることを阻止されている出力ポートの集合を定義する。このフィールドにはN個までのエントリが存在しうる。なお、最初は空である。

【0060】・予約 (Reservations): これは与えられたタイムスロットに対するスケジューリング予約を記録する。CORPSは、現タイムスロット (CTS) に対してこのフィールド内のすべてのエントリがすべてのSMにわたり同一であることを保証する。従って、クロスバモジュールは、任意のSM (CTS) から、セルの現在の入出力スケジューリングを読み出すことができる。アルゴリズムの一貫性チェックは、クロスバコントローラが十分な処理時間を有する場合に、すべてのSMの間でこのフィールドを比較することにより、クロスバモジュールによって実行されることができる。

【0061】CORPSスケジューリングアルゴリズム
各SMは、ここで説明するCORPSスケジューリングアルゴリズムに従う。CORPSには、与えられたSMがどの出力ポートをスケジューリングしようとするかに関する制約はない。どの出力ポートをスケジューリングしようとするかの選択は、各SMにまかされ、そのVOQにサービスする固有のポリシーに従う。図14はCORPSスケジューリングアルゴリズムを示すフローチャートである。以下、タスクボックス101~110について、同図のフローに従いながら説明する。

【0062】まず、タスク101において、前のSMからSメッセージを受信し、各SEに対してTTL (存続時間) をデクリメントする。更に、与えられたSEに対してTTL>0の場合には、TSI (タイムスロットID) をデクリメントし、TSIにおけるメモリアレイSCを更新する。TTL=0の場合には、そのSEをSメッセージから除去する。また、キャリー (CARRY) フラグをCARRY=FALSEにリセットする (タスク109参照)。

【0063】続いて、タスク102 (試行スロットの計算) において、適当な優先マトリクスを用いて、どの将来のタイムスロット (FTS: Future Time Slot) をスケジューリングしようとするかを計算する。簡単のため、そのマトリクスは、 $FTS = f(CTS, SM_ID)$ の形の関数 f にエンコードされうる。

【0064】更に、SMが送信用にどの出力ポート (OPIS) をスケジューリングしようとするかを選択する (タスク103: 出力ポートの選択)。なお、出力ポートを選択する戦略は、前のタスクの結果に依存す

る可能性がある。CORPSはこの戦略を指定しない (例えば、出力ポートの重み付きラウンドロビン選択が使用可能である)。

【0065】続いて、SC (FTS) の予約エントリのうちで、SM-IDがこのスケジューリングを実行するSMに等しいものがあるかどうかを単にチェックする (タスク104: スロットを自分が所有しているかのテスト)。

【0066】もし、SM-IDがこのスケジューリングを実行するSMと異なるならば (タスク104のNO)、更に、SC (FTS) の阻止エントリのうちで、OPI (出力ポートID) が、スケジューリングを試みている出力ポートOPISに等しいものがあるかどうかをチェックする (タスク105: 自分は阻止されているかのテスト)。

【0067】自分が阻止されているならば (タスク105のYES)、Sメッセージを次のSMに渡す (タスク106)。

【0068】自分が阻止されていないならば (タスク105のNO)、SC (FTS) の予約エントリのうちで、OPI (出力ポートID) が、スケジューリングを試みている出力ポートOPISに等しいものがあるかどうかをチェックする (タスク107: そのスロットは取られているかのテスト)。

【0069】そのスロットが取られていない場合には (タスク107のNO)、SC (FTS) に、自己のSM-IDを有しOPIがOPISに等しい予約エントリを作成し、TTL=N、TSI=FTSで、SM-IDは自己のIDに等しく、OPI=OPISであるSEを作成する (タスク108: スロットを取る)。その後、タスク106 (Sメッセージの受け渡し) へ進む。

【0070】このスロットを自分が所有している場合 (タスク104のYES) あるいはそのスロットが既に取られている場合 (タスク107のYES) には、繰越し動作が既に開始されているかどうかを検査し、フラグCARRY=TRUE/FALSEをチェックする。CARRY=TRUEの場合には、SC (FTS) の阻止フィールドに、OPI=OPISのエントリを作成し、そうでない場合にはCARRY=TRUEにセットし、更に、 $FTS = FTS + N$ にセットする (タスク109: 繰越し)。

【0071】タスク109により繰越し動作が実行されると、続いて健全性チェックが行われる (タスク110)。即ち、FTSは、CTSから $2N^2$ より遠くに離れてはならない。 $(FTS - CTS) > 2N^2$ である場合 (タスク110のNOK)、エラーメッセージを出して処理を中止する。 $FTS - CTS \leq 2N^2$ の場合には (タスク110のOK)、タスク104へ戻る。

【0072】CORPSアルゴリズムを用いることで以下の利点が生じる。バックログのあるVOQは、最終的

にはそのSMによって選択されると仮定しても、枯渇することはない。VOQ q がSM p によって選択されると仮定すると、図14によれば、SM p が q をスケジューリングすることに成功せずに予約ループを抜ける唯一の場合は、試みたスロットに対して阻止されている場合である。SM p が阻止されているとは、キュー q が既にスケジューリングされていることを意味するが、以下の点に注意すべきである。ループを抜ける他の唯一の場合があるとすれば、健全性チェックを通る場合であるが、これは、繰越し動作が次のNフレームに空きスロットを見つけないことを意味する。1つのコリジョンにかかわるSMはたかだかN個であり、複数のコリジョンは阻止手続きによって禁止されるため、ループからこのようにして抜けることはない。

【0073】同じ出力ポート q を連続してスケジューリングしようとする m 個の入力ポート(SM)のセットを M とする。さらに、 Δt のタイムスロットの間に出力ポート q に対してSM i によってスケジューリングされるスロットの個数を $n_i^q(\Delta t)$ とする。スケジューラは、任意の期間 Δt 及び $i, j \in M$ に対して、

$$|n_i^q(\Delta t) - n_j^q(\Delta t)| \leq N$$

である場合、 m -フェア(m -公平)であるという。換言すれば、SMは、他のいずれのSMよりも前にN個の予約をすることはできない。

【0074】CORPSは、 m -公平($1 < m \leq N$)である。与えられたスロット t sにおいて出力ポート q に対して m 個のSMが衝突していると仮定する。衝突している m 個のSMはそれぞれ、そのスロットに対して阻止されてはいない。もし阻止されていれば、そのスロットが既に取られているかどうかのテストをすることさえできないからである(図14のタスク105)。これらの m 個のSMがスロット t sにおいて阻止されていない場合、 $t s + nN$ ($1 < n \leq i$)の間に m 個の空きスロットがなければならない。なぜならば、将来にこれらのスロットにアクセスする唯一の方法は繰越し動作を通してであり、しかも、これらのSMはこれらのスロットに対して繰越し動作を実行していないことが分かっている(さもないとそれらは $t s$ において阻止されている)からである。このことは、次の i フレーム以内に、衝突中のSMがそれぞれ q に対してスケジューリング要求をすることになることを意味する。ここで、それらは現フレームのN個の連続するスロットに対して衝突し続け、しかも、各コリジョン(衝突)はSMごとに次の i フレームに1つのスケジューリングを生成するとすれば、各SMは、出力ポート q に対して次の i フレームに全部でNスロット予約することになる。このように、コリジョンが解決される i フレームの i N個のスロットから取られるスロットのいずれのサブセットも、他のSMよりもNスロットより多い利益を有するSMを含むことはできない。

【0075】最後の注意は興味深いものである。それは、たとえ測定期間がどれほど長くても、連続してバックログのあるVOQは、他のSMの対応するVOQの前にN個より多くのパケットをサービスされることはないことを意味するからである。実際、十分長い期間では、すべての衝突するSMは厳密に同数の予約を得ることになる。

【0076】さらに、重い負荷のもとでは、共通の出力ポートを有するキューはすべて、それらのSMによって同じ回数だけ選択されると仮定すれば、すべて同じスロットを有する(図14のタスク108)。

【0077】CORPSアーキテクチャについていくつかのコメントをしておかなければならない。SMどうしの間でスケジューリング情報を渡すために用いられる通信チェーンは、将来の少なくともN個のスロットである限り、任意の方法でスロットのスケジューリングパターンを変更するために使用することが可能である。例えば、出力ポート予約を取り下げることでも可能である。この機能は、SMがコリジョンにより遠い将来に予約をしたばかりであるが、ちょうど次のスロットにおいて、要求するポートが空いたことに気がついた場合に有用となる。SMが、同じパケットに対して別の予約(より近いもの)をする場合、遠いほうの予約は、取下げがなければ帯域の浪費を引き起こす。しかし、予約取下げは、上記の性質に悪影響を及ぼす可能性もある。例えば、衝突したSMが後で予約を取り下げた場合、同じコリジョンにおいてその後にスケジューリングされたパケットの遅延に悪影響を及ぼす。換言すれば、 $i-1$ 個の他のSMと衝突したSMが、この衝突による予約を後で取り下げた場合、システムは、最初に $i-1$ 個のSMのみが衝突したのと同じ状態にはない。このスケジューラは、最初的设计目標を満たしながら、できる限り単純なものである。これにより、最終的な実装に要求されるハードウェアは単純なままであることが保証される。

【0078】CORPSは、複数のフレームにわたりパケットスケジューリングを広げることによって衝突を解決する。従って、他のスケジューラに比べて、平均パケット遅延が大きくなると期待することはもっともである。このため、一様トラフィックのもとでCORPSのパフォーマンスを分析する。最終目標は、繰越し動作がどのくらいパケット遅延に影響するかを評価し、競合するスケジューリングアルゴリズムと比べて、システムから最大利用率を得ることである。

【0079】トラフィック負荷に対するパケット遅延に関してスケジューラのパフォーマンスを評価するために、CORPSの分析モデルを作成する。以下では、簡単のため、次の2つの主要な仮定をする。

【0080】(i)一様トラフィック到着過程、および(ii)各SMによるランダムなVOQキュー選択(図14のタスク103)。

【0081】与えられたSMの、出力ポート n 宛のターゲットVOQキュー $Q_{n,n}$ を定義する。パケットは、強度 p でベルヌーイ過程に従ってあらゆる入力ポートに到着する。具体的には、与えられたスロットにおいて、1つのパケットが1つの入力ポートに到着する確率が p である。さらに、あらゆるパケットは、いずれの出力ポート宛の確率も等しい(仮定 i)。従って、ターゲットVOQキューにおけるパケット到着過程は、パラメータ p/N のベルヌーイ分布を有する。

【0082】VOQ選択に関して、与えられたSMの空でない各キューは、スケジューリングのために等確率で選択される(仮定 i)。従って、任意のVOQに対して、当該VOQが空でなければ、 q は選択される確率である。以下、Chipalkatti等(“Protocols for Optical Star-Coupler Network using WDM,” IEEE Journal on Selected Areas in Communications, Vol. 11, NO. 4, May 1993)に従うと、すべてのVOQの利用率が ρ である場合、1つのSMにおいて期待される空でないVOQキューの個数は $1 + (N-1)\rho$ によって与えられる。

【0083】さらに、 q と密接に関連する別の確率を導入すると便利である。 r を任意のキューがそのスケジューラによって選択される確率とする。 q が当該キューが空でないと仮定しているのに対して、 r にはこの制限がないという点で、 r は q とは異なる。次式が成り立つのを見るのは困難ではない。

$$【0084】 r = \rho q = p/N \quad (1)$$

【0085】 $Q_{n,n}$ のふるまいは以下のようにモデル化することができる。パケット到着間時間は明らかに、パラメータ p/N の幾何分布に従う。先頭パケットは、SMによって選択されるまで待機しなければならない。その選択は、与えられたスロットにおいて確率 q で起こる。選択された後、図14のタスク105に従って、スケジューリングから阻止される可能性がある。与えられたスロットにおいてポート m に対して阻止される確率が P_0^m である場合、先頭パケットがSMによって選択されるまでの待機時間は、パラメータ $s = q(1 - P_0)$ の幾何分布に従う。ここで確率はすべての出力ポートに対して同一であるので、上付き添字 m を落とすことができる。 $Q_{n,n}$ が選択された後、常に予約が将来のタイムスロットにおいて行われ、且つパケットはキューから一種のベル

トコンベヤへと送出されると仮定する。ここでパケットは、予約タイムスロットがやって来るのを待機し、やって来た時点でシステムから出る。

【0086】図15は、 $Q_{n,n}$ キューイングシステムに用いられるモデル全体を示す模式図である。到着パケットはまず $Geo(p/N)/Geo(s)/1$ キューに加わる。パケットは、このキューを出ると、追加遅延 D_{corps} を受ける。これは、CORPSがコリジョンを解決する特定の手法の結果生じる遅延である。これは、無限個のサーバを有するボックスによってモデル化される。

【0087】CORPSを通過するパケットの期待遅延は、 $Geo(p/N)/Geo(s)/1$ に対する期待遅延と、平均遅延 $\langle D_{corps} \rangle$ との和によって与えられる(M. J. Karol, M. G. Hluchyj, S. P. Morgan, “Input Versus Output Queuing on a Space-Division Packet Switch”, IEEE Transactions on Communications, Vol. COM-35, No. 12, pp. 1347-1356, Dec. 1987, 参照)。これは、次のように書くことができる(なお、数式中上付きバーで表記している平均値は、明細書本文中で $\langle \rangle$ で囲んで表記しているものと同一である)。

【0088】

【数1】

$$D = \frac{pS(S-1)}{2N\left(1 - \frac{pS}{N}\right)} + S + \overline{D_{corps}} \quad (2)$$

ただし、 S は、 $Geo(s)$ 時間分布の確率変数である。次に、 $\langle D_{corps} \rangle$ の計算について説明する。

【0089】 $Q_{n,n}$ が選択された(先頭パケットが $Geo(p/N)/Geo(s)/1$ を出た)後、いくつかの事象が起こり得る。まず、SMは、試行しているスロットを所有していないことを確認しなければならない(図14のタスク104)。スロットが、SMによって、出力ポート n に対して所有される確率を P_0^n とする。さらに、与えられたSMが、与えられたタイムスロットにおいて、出力ポート n に対して阻止される確率を P_0^n とする。これから、次式を導くことができる。

【0090】

【数2】

$$P_0^n = 1 + \frac{1}{N} \left[1 + (N+1)(1-r)^N - \frac{2[1 - (1-r)^{N+1}]}{r} \right] \quad (3)$$

【0091】CORPSによれば、SMが訪れているスロットは、このスロットが同じ出力ポートに対して前のコリジョンを解決するために使用されている場合に限り、そのSMが予約しようとするのを阻止することができる。

【0092】その結果、SMがいずれかのポートを所有する確率は、

$$P_0 = 1 - (1 - P_0^n)^N \quad (5)$$

となる。

【0093】図14のタスク104によって生じる期待遅延 $\langle D_0 \rangle$ は次式によって与えられる。

【0094】

【数3】

$$\overline{D_0} = \sum_{k=1}^N NkP_0^k(1-P_0) = N \left[\frac{P_0 - P_0^{N+2}}{1-P_0} - (N+1)P_0^{N+1} \right] \quad (6)$$

【0095】SM_mが最初に訪れたスロットが空いている場合（図14のタスク104、105、および107のテストがすべてNO）、優先マトリクス方式が使用されていることにより、バケットの平均遅延〈D_{corps}〉がNになることを見るのは容易である。〈D_{corps}〉>Nで、コリジョンがない場合、少なくとも1つの予約が、将来の第2のフレームへとこぼれる。ここで、コリジョンによって受ける遅延D_cについて調べる。特定のスロットに対してi-1個の他のSMとのコリジョンが

$$P[D_{corps} = jN | v = 1] = \begin{cases} 1 & j = i \text{ の場合} \\ 0 & \text{その他} \end{cases} \quad (7)$$

となる。

【0097】上記の式は単に、SM_mがスロットを訪れる最初のSMである場合、そのバケットはNスロット遅延されるということである。コリジョンが起こらない場合、CORPSスケジューラは将来の1フレームをスケジューリングするからである。次に、SM_s (s≠m)の任意の出力ポート（特に出力ポートn）に対するVO

$$P[D_{corps} = jN | v = i] = \begin{cases} 0 & j > i \text{ の場合} \\ \binom{i-1}{j-1} r^{j-1} (1-r)^{i-j} & \text{その他} \end{cases} \quad (8)$$

【0099】式(8)の上段は、mがスロットを訪れるi番目のSMである場合、その遅延はただだかiNであるということである。下段の二項係数は、i-1個のSMがmの前にスロットを訪れた場合、これらのうちのj-1個のSMがmと衝突する可能性があるということである。(D_{corps}=jNかつv=i)の形の事象の同時分布は、上記の表式に1/Nを乗じることによって容易に導出することができる。なぜなら、SM_mは、スロット1≤i≤Nのi番目の訪問者であることが等しく確からしいからである（図7参照）。

【0100】次に、1つのバケットの期待遅延D_cは次のように導出することができる。

【0101】

【数6】

$$P_0^N = 1 + \frac{1}{N} \left[1 + (N+1)(1-r)^N - \frac{2[1 - (1-r)^{N+1}]}{r} + 2 + r - \frac{(1-r^{N+1})}{1-r} \right] \quad (11)$$

【0106】Geo(s)に対して、〈S〉=1/sで、〈S(S-1)〉=2(1-s)/s²であることに注意すると、1つのバケットがシステムで受ける全平

起こる場合、そのスロットに関してSM_mが有する優先順位に依存して、遅延D_cはNからiNまでの間で変わりうる。そこで、SM_mがそのスロットを訪れるi番目のSMである場合にバケット遅延がjNである確率をP[D_{corps}=jN | v=i]とする。例えば、mがそのスロットを訪れる最初のSMである場合、

【0096】

【数4】

Qキューが空でなく、かつ、sによって選択される確率はrであることを想起すると、P[D_{corps}=jN | v=i]に対する一般式が次のようになることを見るのは困難ではない。

【0098】

【数5】

$$\overline{D_c} = \frac{N(N-1)}{2} r + N \quad (9)$$

【0102】CORPSスケジューラにより生じる全遅延は次のようになる。

【0103】

【数7】

$$\overline{D_{corps}} = \overline{D_0} + \overline{D_c} \quad (10)$$

【0104】計算すべき最後の確率はP₀である。これは、与えられた出力ポートnに対して、与えられたスロットにおいて、1つのSMが阻止される確率である。次式を示すことができる。

【0105】

【数8】

均遅延は次のようになる。

【0107】

【数9】

$$D = \frac{p(1-s)}{Ns^2\left(1-\frac{p}{Ns}\right)} + \frac{1}{s} + N \left[\frac{P_0 - P_0^{N+2}}{1-P_0} - (N+1)P_0^{N+1} \right] + \frac{N(N-1)}{2}r + N \quad (12)$$

ただし、 $s = q(1-P_0)$ である。最初の3項は、スケジューリングが行われる前の、VOQキューにおける遅延に対応する。第3項は、CORPSのパイプラインおよびコリジョン解決機能により、パケットが待機するのに必要な追加時間に対応する。

【0108】図16に、CORPSの遅延対スループットの解析的結果を、CORPSスケジューラを備えた16×16スイッチのシミュレーションと比較したものを示す。この図において、パイプラインおよびコリジョン解決方式が使用されることにより、パケットがSMスケジューラによって選択されるまでに受ける平均キューイング遅延と、CORPS遅延との間に違いがある。図から分かるように、解析的予測は、シミュレートされたシステムのふるまいと良く一致する。

【0109】この図は、負荷のすべての範囲を通じて、スケジューリング遅延がキューイング遅延よりも優勢であることを示している。非常に高い負荷の場合（キューが形成され始めるとき）にのみ、キューイング遅延が重要になる。これは、パケットがVOQキューに到着するとすぐに将来のパケットをスケジューリングすることにおいて、CORPSがうまくはたっていることを意味する。他方、CORPSによって生じる平均遅延は、軽負荷の場合のおよそ1フレームから、負荷が0.85に達するときの約5フレームまで、増大する。

【0110】完全を期するため、図17に、16×16 CORPSスイッチにおける全遅延の相補分布を示す。曲線は、シミュレーションによって得られた、負荷が0.8および0.85の場合のものである。まず、いずれのパケットも、システムを通過するのに N^2 スロットより多くはかからないことに注目される。これは、CORPSでは多重コリジョンが起こることを許していないことによる。実際、分布のテールは、 $N^2/2 = 128$ 付近のあたりで終わっているように見える。しかし、システムが非常に大きい負荷によって駆動される場合、パ

ケット遅延は N^2 に近づくようである。

【0111】図18は、CORPSを実現するシステムブロック図である。VOQMモジュールは、パケットを仮想出力キューVOQに入れる。また、このモジュールは、与えられたキューに代わって、SMモジュールに対して要求を行う。SMモジュールは、メッセージ受渡しを制御し、CORPSスケジューラを実現する。SMモジュールは、VOQMと通信して、将来のスロット予約について通知する。この通知はVOQMに保持され、与えられたスロットにおいて、パケットが、交換されるべきクロスバレジスタに転送されるようにする。

【0112】図中、SMとクロスバコントローラの間の通信はバスを通じて行われるように示されているが、この特定の種類の通信である必要はない。

【0113】スケジューリングアルゴリズムどうしの公平な比較では、平均遅延やスループットのようなパフォーマンス尺度のみならず、複雑さおよび実装コストも考慮すべきである。第1の選択基準は高いスループットである。さらに、VOQで動作するスケジューラのみを比較する。そこで、本発明と競合するスケジューラとして、1-SLIP及びRRGSとの比較を行う。複数回のイテレーションではなく1イテレーションのSLIPを選択する理由は、比較プロセスの公平性のためである。すなわち、任意の入力ポートにおいて、スロット当たりたかだか1回の決定をすることができると仮定する。 i -SLIP ($i > 1$)は、実質的に、スロット当たり複数回のスケジューリング決定を要求することになる。

【0114】パフォーマンス比較において、解析的結果およびシミュレーション結果の両方をもとにする。一様トラフィックに対するRRGSおよびSLIPの遅延パフォーマンスは次のように近似することができる。

【0115】

【致10】

$$\overline{D}_{RRGS} = \frac{p(1-q)}{Nq^2\left(1-\frac{p}{Nq}\right)} + \frac{1}{q} + \frac{N}{2} \quad (13)$$

$$\overline{D}_{SLIP} = \frac{pN}{2(1-p)} \quad (14)$$

【0116】RRGSの結果については、本出願人による特願平11-172584号に記載されており、SLIPの結果については、N. McKeown, "Scheduling Cell sinan Input-Queued Switch", PhD Thesis, Universit

y of California at Berkeley, 1995、に記載されている。

【0117】図19に、これらのアルゴリズムの平均遅延対スループットのパフォーマンスを、CORPSと対

照して示す。この図から明らかなように、RRGS及びCORPSは、遅延が大きくなる前には、SLIPよりもずっと高い負荷にたえることができる。容易に分かるように、これらの曲線の微分は、高負荷の場合、RRGS及びCORPSのほうがかなり小さい。しかし、いずれのアルゴリズムも、中程度から軽い負荷ではオフセット遅延バジェットを有する。RRGSの場合、これは、パイプライン法が使用されていることのみによるものである。CORPSの場合、既に説明したように、追加遅延はコリジョン解決によるものである。しかし、CORPSは、RRGSに比べて2つの利点を有する。

(i) SMがどの出力ポートを選択するかについて選択の自由がある。

(ii) 厳密に公平なスケジューラである。

SLIPもまた公平なスケジューラであるが、そのコリジョン解決プロセスは、CORPSのものとは全く異なる。

【0118】前述のように、CORPSは、どの出力ポートにスケジューリングを試みるかについて完全な選択の自由を与える。すなわち、各VOQMは、与えられたVOQに代わって、スケジューリングされる出力ポートを自由に選択することができる。このことは、スケジューラ設計ストラテジの重要な部分であった。従って、多くのアルゴリズムが、CORPSとともに、VOQ選択に使用可能である。これまで、そのようなアルゴリズムの1つ、すなわち、空でないVOQのうちのランダム選択について説明した。他のVOQ選択ストラテジの例も可能である。VOQ選択ストラテジは、協調的選択ストラテジおよび非協調的選択ストラテジという2つのクラスに分類することができる。

【0119】非協調的VOQ選択ストラテジは、VOQ選択決定が、他の入力ポートとは独立に、入力ポート(VOQM)ごとに行われるものである。CORPSの分析に用いたランダム選択ストラテジはこのクラスに属する。

【0120】重み付き公平キューイング(WFQ: Weighted Fair Queuing)は、パケット交換研究文献において広く知られたサービスストラテジである(例えば、H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks", In Proceedings of IEEE, Vol. 83, no. 10, pp. 1374-1396, Oct. 1995, 参照)。その考え方は、所定の重みに従って、出力リンク容量に対して競合する複数のキューのサービスレートを規制するというものである。VOQ CORPSスイッチにおいて、出力ポート帯域は、ある種の呼受け付けコントローラによって複数のVOQMに分割することができる。その場合に、WFQを用いて、VOQキューの最大サービスレートが、与えられた出力ポートのVOQM帯域分を超えないように強制することができる。

【0121】レート制御サービス(RCS: Rate-Controlled Service)規律は、与えられたトラフィックフローが、ネットワークエントリポイントでいくつかのバースト性制約を満たすと仮定する(L. Georgiadis, R. Guerin, V. Pens, "Efficient Network QoS Provisioning Based on per Node Traffic Shaping", Proceedings of INFOCOM'96, vol. 1, pp. 102-110, 1996, 参照)。これらの制約は一般に、ネットワークのエッジにおけるトラフィックシェーパによって強制される。さらに、トラフィックシェーパは、中間スイッチにも配置され、トラフィックが、ネットワーク内の各中間交換ポイントでそれらの制約に従うようにされる。トラフィックシェーパは一般に、リーキーバケットアルゴリズムによって実現される。J. Turner, "New Directions in Communications, or Which Way to the Information Age?", IEEE Communications Magazine, Vol. 24, pp. 8-15, 1986, には、そのようなアルゴリズムの1つが記載されている。基本的なリーキーバケットは、2つのキュー(1つはデータ用で、1つはトークンすなわちパーミット用)を有するシステムである。キュー上のデータパケットは、サービスを受けるためにはパーミットを必要とする。制限された個数のパーミットのみがパーミットキューに格納される。パーミットは、一定レートで生成される。この種のトラフィックシェーパは、VOQのうちのいずれがサービスを受けるかを規制するために使用可能である。適格なVOQのうちからは、キュー選択に任意のアルゴリズムを用いることが可能である。

【0122】上記の2つのサービス規律は、パケットネットワークにおけるサービス品質(QoS)のサポートに使用可能であり、それ自体、活発な研究分野である。このようなQoSサポートストラテジは、非協調的なタイプのものになることが多い。それは、他のトラフィックストリームとは無関係に、VOQの予測されるサービス挙動を保証することになるからである。このクラスに属するアルゴリズムは、ビデオや音声ストリームのような、厳しいQoSアプリケーションをサポートするスイッチで使用可能である。

【0123】協調的VOQ選択ストラテジは、VOQ選択がスイッチ内のVOQのセット全体の状態に依存するような選択ストラテジである。このストラテジは一般に、各フローのサービスに集中するよりも、最大スループットのようなスイッチ全体のふるまいを良くすることを目標とする。従って、このようなストラテジをスイッチで使用するのには、QoS要求条件に対する約束なしに、データトラフィックをサポートする場合である。

【0124】協調的ストラテジの場合、他のVOQの状態のような追加情報をCORPSスケジューラに提供する必要があります。キューの状態に関する情報は常に「古い」ため、サービスストラテジは、古い情報に関してロバストでなければならない。

【0125】最大マッチング問題とは、与えられたグラフの辺のうちから、グラフの頂点の対をつなぐ辺で、対の総数を最大にするような辺のサブセットを求める問題である (Cormen, Leiserson and Rivest, "Introduction to Algorithms", McGraw-Hill, 1990、参照)。しかし、どの頂点も、つなぐ選択された辺を複数本有することはできない。あらゆるスロットで交換されるバケットの個数を最大にする場合、最大2部マッチング (MBM: Maximum Bipartite Matching) 問題を解く必要がある (R. E. Tarjan, "Data Structures and Network Algorithms", Society for Industrial and Applied Mathematics, Pennsylvania, Nov. 1983、参照)。適当な計算量でMBMを解くアルゴリズムが利用可能である (J. E. Hopcroft, R. M. Karp, "An $n^{5/2}$ Algorithm for Maximum Matching in Bipartite Graphs", Society for Industrial and Applied Mathematics J. Comput., 2 (1973), pp. 225-231、参照)。本発明では、VOQが空きであるか否かの状態情報は、通信チェーンを通じて送られて、VOQMに渡される。ここで、MBMアルゴリズムは、次フレームのスロットにどのキューがサービスするかを決定する。興味深い点であるが、CORPSによれば、MBMアルゴリズムによって選択されないキューも、将来の予約を試みる事が可能である。

【0126】最大重み2部マッチング (MWBM: Maximum Weight Bipartite Matching) 問題は、上記のMBM問題と類似している。主な相違点は、前者では、重みがグラフの辺に関連づけられ、目的は、マッチングの辺の重みの総和を最大にする辺のセットを求めることである。他の研究者は、MWBMアルゴリズムを用いると、非一様トラフィックのもとでは、スループットに関してMBMストラテジよりパフォーマンスが優れていることを示している (N. McKeown, V. Anantharam, J. Walrand, "Achieving 100% Throughput in an Input-Queued Switch", Proceedings of Infocom'96, San Francisco, March 1996、参照)。考え方は、非一様トラフィックの場合を扱うために、VOQキューサイズを重みとして用いることである。

【0127】また、上記文献によれば、入力トラフィックが受け付け可能である限り、MWBMアルゴリズムは安定である、すなわち、VOQキューは爆発しない。あらゆる出力ポートに対して、1個の出力ポートへの入力トラフィックレートの総和がその容量を超えない場合に、トラフィックが受け付け可能であるという。この興味深い結果は、MWBMの安定性は、古い情報の存在下でも、すなわち、重みがいくつかの過去のタイムスロットのキューレベルに基づいていても、維持されるということである。この場合も、VOQのキューレベル情報はすべてのVOQMに渡され、出力ポートに対する要求がSMへと発行される前に、MWBMアルゴリズムが各モジュールで実行されるようにすることができる。

【0128】

【発明の効果】以上詳細に説明したように、本発明によれば、まず、時間軸をフレーム化して優先マトリクスを用いた予約を行うことにより、タイムスロット巡回順序を規則的なものとし、実装及び制御を容易にするという効果がある。また、RRGSとは異なり、スケジューリングの規則をSM個数の偶奇によって変える必要がなく、この点でも実装及び制御が簡単であるといえる。

【0129】さらに、本発明による繰越しラウンドロビンパイプラインスケジューラ (CORPS) によれば、クロスバ高速スイッチファブリックの入線間での公平なスケジューリングが可能となる。CORPSは、将来のスロットのバケットをスケジューリングすることにより、ラインごとスロットごとに1つのスケジューリング決定を行う。スケジューリングされるキューの選択は任意であるため、トラフィックのサービス品質をサポートすることに適している。CORPSは、出力ポート間の競合を公平に解決する。

【0130】さらに他の効果および変形を考えることは当業者には容易であって、本発明は、ここで説明した具体例に限定されない。特許請求の範囲に記載した本発明の構成の技術思想あるいは技術的範囲から離れることなく、さまざまな変形例を考えることが可能である。

【図面の簡単な説明】

【図1】集中VOQスケジューラを示す概略的ブロック図である。

【図2】(A)は並列方式の、(B)はラウンドロビン方式のアーキテクチャをそれぞれ示す分散スケジューラアーキテクチャの模式図である。

【図3】入力バッファ型スイッチアーキテクチャの説明図である。

【図4】入力ポート分散スケジューラの構成を例示する概略的ブロック図である。

【図5】4×4クロスバスイッチを用いた場合のRRGSによるパイプラインスケジューリング決定の一例を示すタイムチャートである。

【図6】図5を個々のSMのタイムスロット巡回順序に着目して表現したタイムチャートである。

【図7】本発明によるスケジューラの一実施形態におけるコリジョンを解決するのに用いられる優先マトリクスの一例 (ポート数N=4) を示す説明図である。

【図8】本発明によるスケジューラの一実施形態におけるパイプラインスケジューリング決定の一例 (ポート数N=4) を示す説明図である。

【図9】本発明によるスケジューラの一実施形態におけるコリジョンを解決するのに用いられる優先マトリクスの一例 (ポート数N=5) を示す説明図である。

【図10】本発明によるスケジューラの一実施形態におけるパイプラインスケジューリング決定の一例 (ポート数N=5) を示す説明図である。

【図11】本発明によるスケジューラの一実施形態におけるSM間の繰越し動作を示す説明図である。

【図12】本実施形態におけるSメッセージのフォーマット図である。

【図13】本実施形態におけるスケジューラモジュールのデータ構造体のフォーマット図である。

【図14】本実施形態におけるCORPSスケジューリングアルゴリズムを示すフローチャートである。

【図15】CORPS VOQキューイングモデルを示す

す模式図である。

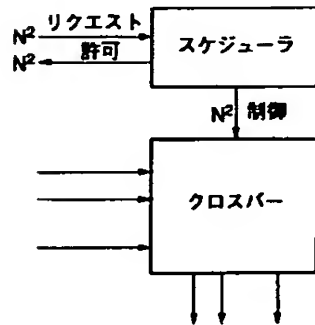
【図16】システム負荷の関数としてパケット遅延を表すグラフである。

【図17】CORPSスケジューラを備えた16×16スイッチの相補的遅延分散を示すグラフである。

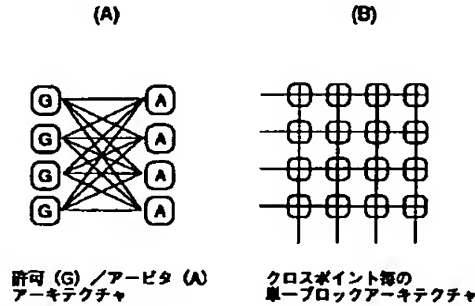
【図18】CORPSコントローラの一例を示すブロック図である。

【図19】さまざまな競合スケジューラの、システム負荷に対する期待遅延を示すグラフである。

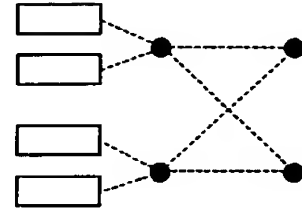
【図1】



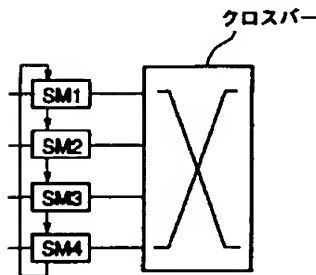
【図2】



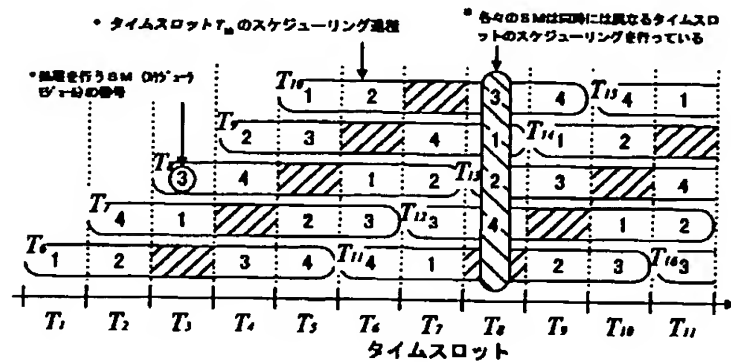
【図3】



【図4】



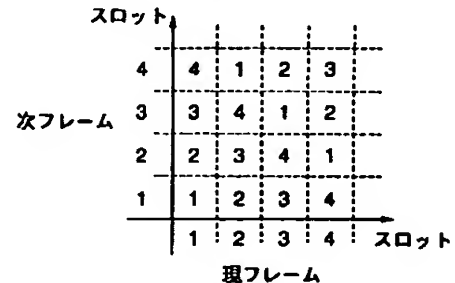
【図5】



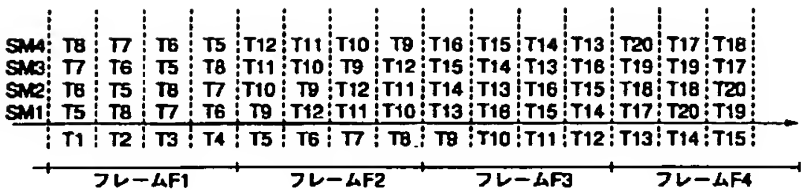
【図6】

タイムスロット スケジューリングモジュール	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
SM 1	T6	T4	T7	T5	T10	T8	T11	T9	T14	T12	T15
SM 2	T3	T6	T4	T9	T7	T10	T8	T13	T11	T14	T12
SM 3	T5	T3	T8	T6	T9	T7	T12	T10	T13	T11	T16
SM 4	T2	T7	T5	T8	T6	T11	T9	T12	T10	T15	T13

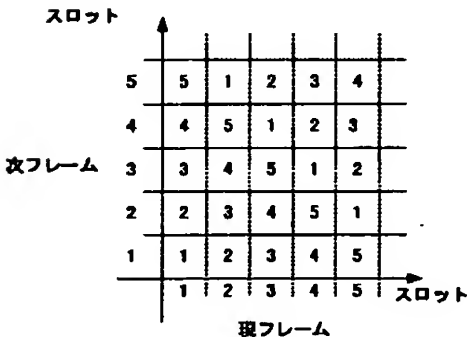
【図7】



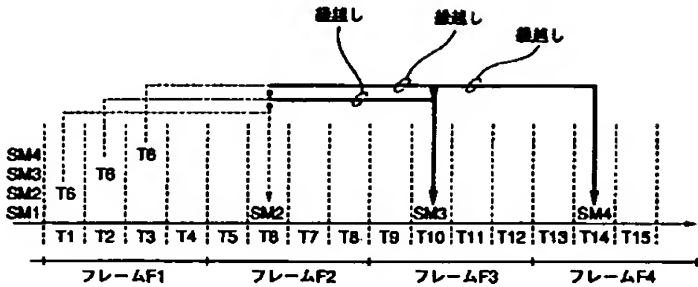
【図8】



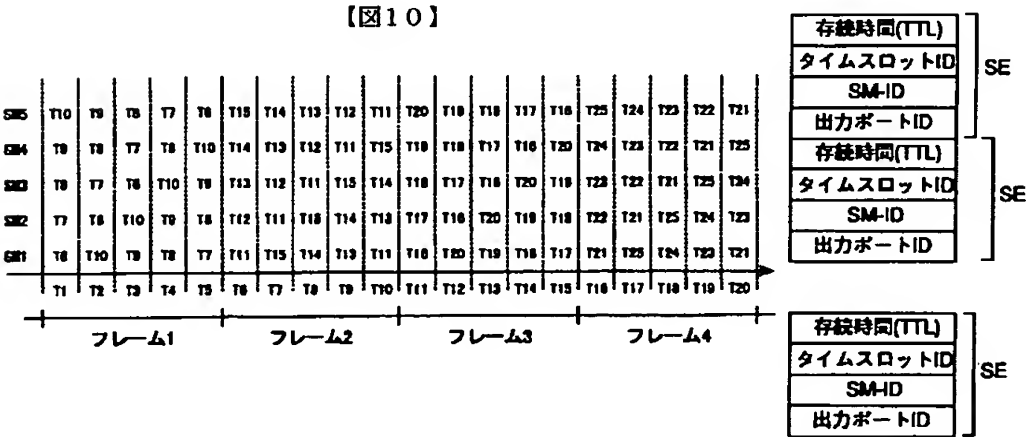
【図9】



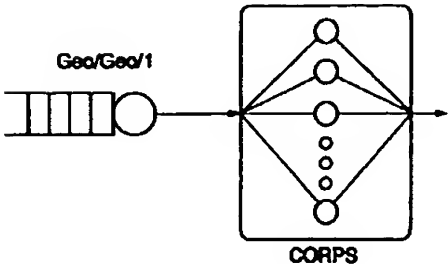
【図11】



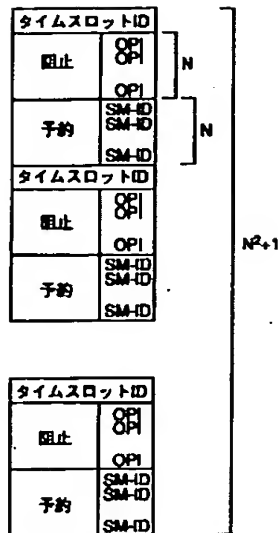
【図12】



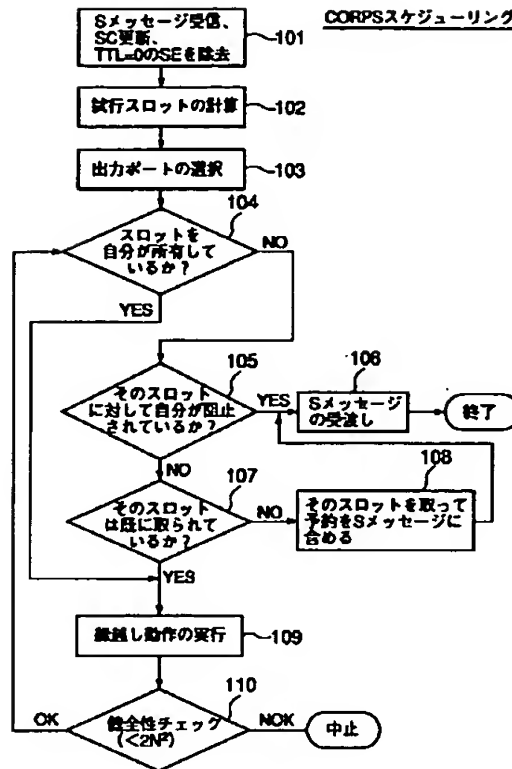
【図15】



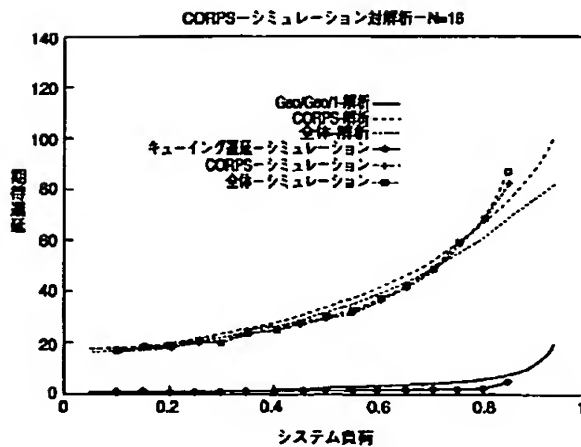
【図13】



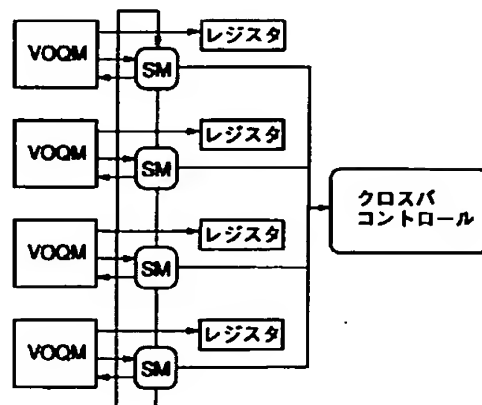
【図14】



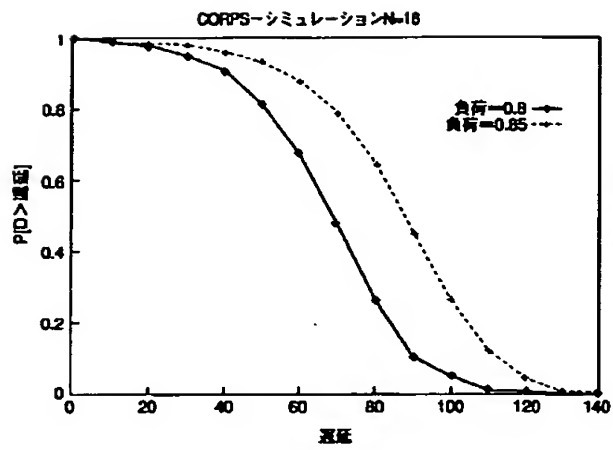
【図16】



【図18】



【図17】



【図19】

